	Application No.	Applicant(s)
Notice of Allowability	10/719,443	SUBRAMONEY ET AL.
	Examiner	Art Unit	
	Srirama Channavajjala	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *10/10/06*.

2. ☐ The allowed claim(s) is/are *1-8,10-12,14,16,18-20,25-32,34-36 and 38*. ·

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some*   c) ☐ None   of the:

        1. ☐ Certified copies of the priority documents have been received.

        2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

        3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

    * Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application. **THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.

    (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached

        1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____ .

    (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

    Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL. ·

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)

2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)

3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____

4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material

5. ☐ Notice of Informal Patent Application

6. ☒ Interview Summary (PTO-413), Paper No./Mail Date *11/1/06* .

7. ☒ Examiner's Amendment/Comment

8. ☒ Examiner's Statement of Reasons for Allowance

9. ☐ Other _____ .

DETAILED ACTION

1.   Claims 1-8,10-12,14,16,18-20,25-32,34-36,38 allowed.

2.   Examiner acknowledges applicant's amendment filed on 10/10/2006.

3.   Claims 22-24 have been cancelled [10/10/2006].

Drawings

4.   The Drawings filed on 11/21/2003 are <u>acceptable</u> for examination purpose only.

Double Patenting

5.   In view of applicant's "terminal disclaimer" filed on 10/10/2006, the "double

patent" rejection as set forth in the previous office action is hereby withdrawn.

Interview:

6.   Applicant's Attorney Guojun Zhou, Regd No. 56,478 is thanked for the

telephone interview on 01 November 2006. During that telephone Guojun Zhou

granted authorization to *amend claims 1,5,12,16,18,19,25,29,36* and *cancel*

claims 9,13,15,17,33,37,39.

EXAMINER'S AMENDMENT

7.   An examiner's amendment to the record appears below. Should the changes

and/or additions be unacceptable to applicant, an amendment may be filed as provided

by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be

submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview

with Applicant's Attorney Guojun Zhou, Regd No. 56,478, on 01 November 2006.

The application has been amended as follows:

In the Claims

1. (Currently Amended) A method for performing concurrent mark-sweep

garbage collection, comprising:

receiving an application, the application including a software application to run

concurrently with mark-sweep garbage collection;

executing the application in at least one thread, the application using space in a

heap while being executed;

determining if available space in a heap in the heap falls below a threshold;

performing mark-sweep garbage collection, concurrently while executing the

application, in a heap block of the heap using a first bit vector, a second bit vector, a

mark bit vector pointer, and a sweep bit vector pointer in the heap block, if the available

space falls below the threshold; and otherwise,

continuing executing the application and monitoring if the available space in the
heap falls below the threshold, until the execution of the application is complete;

wherein performing mark-sweep garbage collection includes invoking at least
one garbage collection thread to trace and mark live objects in the heap block
concurrently while executing the application; and wherein the mark bit vector pointer
initially points to the first bit vector, the sweep bit vector pointer initially points to the
second bit vector, and the first bit vector and the second bit vector are toggled with each
other after a marking phase of the mark-sweep garbage collection completes.

2. (Original) The method of claim 1, wherein the heap comprises at least one
heap block.

3. (Original) The method of claim 1, further comprising initializing a concurrent
mark-sweep garbage collector.

4. (Original) The method of claim 3, wherein initializing the concurrent mark-
sweep garbage collector comprises setting each bit in the first bit vector and the second
bit vector to 0, and pointing the mark bit vector pointer to the first bit vector and the
sweep bit vector pointer to the second bit vector.

5. (Currently Amended) The method of claim 1, wherein performing mark-sweep
garbage collection further comprises:

~~invoking at least one garbage collection thread to trace live objects in the heap block concurrently while executing the application; and~~

reclaiming storage space occupied by objects other than the live objects in the block concurrently while tracing the live objects in the block and executing the application.

6. (Original) The method of claim 5, wherein tracing the live objects in the heap block comprises parallel marking the live objects by at least one garbage collection thread.

7. (Original) The method of claim 6, wherein parallel marking the live objects comprises setting bits corresponding to starting addresses of the live objects in a bit vector pointed to by the mark bit vector pointer to 1, by the at least one garbage collection thread.

8. (Original) The method of claim 5, wherein reclaiming the storage space occupied by objects other than the live objects in the heap block comprises sweeping the heap block to make the said storage space allocable by using a bit vector pointed to by the sweep bit vector pointer.

9. (Cancelled)

10. (Original) The method of claim 8, further comprising setting the bit vector back to 0 after completing sweeping the heap block.

11. (Original) The method of claim 1, further comprising performing another cycle of concurrent mark-sweep garbage collection when available space in the heap falls below the threshold again.

12. (Currently Amended) A method for automatically collecting garbage objects, comprising:

receiving a first code, the first code being neutral to a computing platform;

compiling the first code into a second code, the second code being native to the computing platform;

executing the second code in at least one thread by the computing platform; and

automatically performing mark-sweep garbage collection using bit vector toggling, concurrently with the executing second code, to ensure there is storage space available for executing the second code;

wherein automatically performing mark-sweep garbage collection detects if available space in a heap falls below a threshold and invokes the concurrent mark-sweep garbage collection when the available space falls below the threshold by using a first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector pointer for a heap, the mark bit vector pointer initially pointing to the first bit vector, the sweep bit vector pointer initially pointing to the second bit vector, and the first bit vector

and the second bit vector being toggled with each other after a marking phase of the

mark-sweep garbage collection completes.

13. (Cancelled)

14. (Original) The method of claim ~~13,~~12 wherein the heap comprises at least

one heap block.

15. (Cancelled)

16. (Currently Amended) A system for concurrent mark-sweep garbage

collection, comprising:

a root set enumeration mechanism to enumerate references to live objects in a

heap;

a live object tracing mechanism to parallel trace live objects in a heap block and

mark the live objects in a first bit vector pointed to by a mark bit vector pointer in the

heap block, concurrently with execution of an ~~application; and~~ application, the

application using space in the heap while being executed, the live object tracing

mechanism including a live object search mechanism to parallel search live objects in

the heap block by at least one garbage collection thread, and a live object marking

mechanism to parallel mark the live objects in a bit vector stored in the heap block by

the at least one garbage collection thread;

a garbage sweeping mechanism to sweep storage space occupied by garbage objects to make the storage space allocable using a second bit vector pointed to by a sweep bit vector pointer in the heap block, concurrently with the execution of the application and live object marking; and

a bit vector toggling mechanism to toggle the first bit vector pointed to by the mark bit vector pointer and the second bit vector pointed to by the sweep bit vector pointer in the heap block.

17. (Cancelled)

18. (Currently Amended) The system of claim 16, wherein the live object tracing mechanism further comprises:

~~a live object search mechanism to parallel search live objects in a heap block by at least one garbage collection thread;~~

~~a live object marking mechanism to parallel mark the live objects in a bit vector stored in the heap block by the at least one garbage collection thread;~~

a live object scanning mechanism to parallel scan any objects reachable from the live objects in the heap; and

a conflict prevention mechanism to prevent more than one garbage collection thread from marking the same object.

19. (Currently Amended) A ~~managed runtime~~ system for managing memory usage during run-time, comprising:

a just-in-time compiler to compile an application into a code native to an

underlying computing platform;

a virtual machine to execute the application, the application using space in a

heap while being executed; and

a garbage collector to trace live objects, mark the live objects in a first bit vector

pointed to by a mark bit vector pointer in a heap block of a heap of the heap, and toggle

the bit first vector pointed to by the mark bit vector pointer with a second bit vector

pointed to by a sweep bit vector pointer at the end of marking phase, concurrently with

execution of the application, the garbage collector including a live object marking

mechanism to parallel mark the live objects in the first bit vector pointed to by the mark

bit vector in the heap block of the heap, and a bit vector toggling mechanism to toggle

the first bit vector pointed to by the mark bit vector pointer and the second bit vector

pointed to by the sweep bit vector pointer.

20. (Original) The system of claim 19, further comprising a garbage sweeping

mechanism to sweep storage space occupied by garbage objects to make the storage

space allocable using a bit vector pointed to by the sweep bit vector pointer,

concurrently with the execution of the application and live objects marking.

21-24. (Cancelled)

25. (Currently Amended) An article comprising: a machine accessible medium having content stored thereon, wherein when the content is accessed by a processor, the content provides for performing concurrent mark-sweep garbage collection by:

receiving an application, the application including a software application to run concurrently with mark-sweep garbage collection;

executing the application in at least one thread, the application using space in a heap while being executed;

determining if available space in a heap in the heap falls below a threshold;

performing mark-sweep garbage collection, concurrently while executing the application, in a heap block of the heap using a first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector pointer in the heap block, if the available space falls below the threshold; and otherwise,

continuing executing the application and monitoring if the available space in the heap falls below the threshold, until the execution of the application is complete;

wherein performing mark-sweep garbage collection includes invoking at least one garbage collection thread to trace and mark live objects in the heap block concurrently while executing the application; and wherein the mark bit vector pointer initially points to the first bit vector, the sweep bit vector pointer initially points to the second bit vector, and the first bit vector and the second bit vector are toggled with each other after a marking phase of the mark-sweep garbage collection completes.

26. (Original) The article of claim 25, wherein the heap comprises at least one heap block.

27. (Original) The article of claim 25, further comprising content for initializing a concurrent mark-sweep garbage collector.

28. (Original) The article of claim 27, wherein content for initializing the concurrent mark-sweep garbage collector comprises content for setting each bit in the first bit vector and the second bit vector to 0, and pointing the mark bit vector pointer to the first bit vector and the sweep bit vector pointer to the second bit vector.

29. (Currently Amended) The article of claim 25, wherein content for performing mark-sweep garbage collection further comprises content for:

~~invoking at least one garbage collection thread to trace live objects in the heap block concurrently while executing the application; and~~

reclaiming storage space occupied by objects other than the live objects in the block concurrently while tracing the live objects in the block and executing the application.30. (Original) The article of claim 29, wherein content for tracing the live objects in the heap block comprises content for parallel marking the live objects by at least one garbage collection thread.

31. (Original) The article of claim 30, wherein content parallel marking the live objects comprises content for setting bits corresponding to starting addresses of the live objects in a bit vector pointed to by the mark bit vector pointer to 1, by the at least one garbage collection thread.

32. (Original) The article of claim 29, wherein content for reclaiming the storage space occupied by objects other than the live objects in the heap block comprises content sweeping the heap block to make the said storage space allocable by using a bit vector pointed to by the sweep bit vector pointer.

33. (Cancelled)

34. (Original) The article of claim 32, further comprising content for setting the bit vector back to 0 after completing sweeping the heap block.

35. (Original) The article of claim 25, further comprising content performing another cycle of concurrent mark-sweep garbage collection when available space in the heap falls below the threshold again.

36. (Currently Amended) An article comprising: a machine accessible medium having content stored thereon, wherein when the content is accessed by a processor, the content provides for automatically collecting garbage objects by:

receiving a first code, the first code being neutral to a computing platform;

compiling the first code into a second code, the second code being native to the

computing platform;

executing the second code in at least one thread by the computing platform; and

automatically performing mark-sweep garbage collection using bit vector

toggling, concurrently with the executing second code, to ensure there is storage space

available for executing the second code;

wherein automatically performing mark-sweep garbage collection detects if

available space in a heap falls below a threshold and invokes the concurrent mark-

sweep garbage collection when the available space falls below the threshold by using a

first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector

pointer for a heap, the mark bit vector pointer initially pointing to the first bit vector, the

sweep bit vector pointer initially pointing to the second bit vector, and the first bit vector

and the second bit vector being toggled with each other after a marking phase of the

mark-sweep garbage collection completes.

37. (Cancelled)

38. (Original) The article of claim 37 36, wherein the heap comprises at least one

heap block.

39. (Cancelled)

In the Title

Pursuant to MPEP 606.01 the Title is changed to read

--METHOD AND SYSTEM PERFORMING CONCURRENTLY MARK-SWEEP GARBAGE COLLECTION INVOKING GARBAGE COLLECTION THREAD TO TRACK AND MARK LIVE OBJECTS IN HEAP BLOCK USING BIT VECTOR —

Reasons for allowance

Claims *1-8,10-12,14,16,18-20,25-32,34-36,38* are allowed

The following is an examiner's statement of reasons for indication of allowable subject matter: The prior art of record does not disclose, make obvious, or otherwise suggest the structure of the applicant's "*wherein performing mark-sweep garbage collection includes invoking at least one garbage collection thread to trace and mark live objects in the heap block concurrently while executing the application; and wherein the mark bit vector pointer initially points to the first bit vector, the sweep bit vector pointer initially points to the second bit vector, and the first bit vector and the second bit vector are toggled with each other after a marking phase of the mark-sweep garbage collection completes*", claim 1,25;

"*wherein automatically performing mark-sweep garbage collection detects if available space in a heap falls below a threshold and invokes the concurrent mark-sweep garbage collection when the available space falls below the threshold by using a first bit vector, a second bit vector, a mark bit vector pointer, and a sweep bit vector pointer for a heap, the mark bit vector pointer initially pointing to the first bit vector, the sweep bit vector pointer initially pointing to the second bit vector, and the first bit vector and the second bit vector being toggled with each other after a marking phase of the mark-sweep garbage collection completes*", in claim 12,36;

"*a live object marking mechanism to parallel mark the live objects in a bit vector stored in the heap block by the at least one garbage collection thread;*

a bit vector toggling mechanism to toggle the first bit vector pointed to by the

mark bit vector pointer and the second bit vector pointed to by the sweep bit vector

pointer in the heap block", in claim 16;

" a live object marking mechanism to parallel mark the live objects in the first bit

vector pointed to by the mark bit vector in the heap block of the heap, and a bit vector

toggling mechanism to toggle the first bit vector pointed to by the mark bit vector pointer

and the second bit vector pointed to by the sweep bit vector pointer", in claim 19.

These features, together with the other limitations of the independent claims

are novel and non-obvious over the prior art of record. The dependent claims 2-8,10-

11,14,18,20,26-32,34-35,38 being definite, enabled by the specification, and further

limiting to the independent claims are also allowable.

The newly cited reference Cohn,Joesph A et al. EP 1349077 published on

October 2003, is directed to optimization of memory usage based on garbage collection

simulation, more specifically memory usage data is received wherein the memory usage

data comprises timing information [see Abstract],, particularly, timing information as

captured in the memory usage data allows for plotting various information versus the

overall time of the application's execution [page4, col 6, 0026]. Cohn also suggests

heap parameter input view comprises a JVM selector . A user can change the JVM

version/type options that are necessary for re-running their application by selecting a

different JVM version [page 5, col 8, 0036].

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance".

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alam, Hosain, T, can be reached on (571) 272-3978. The fax phone numbers for the organization where the application or proceeding is assigned is 571-273-8300 Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free)

sc
Patent Examiner.
November 1, 2006.   SRIRAMA CHANNAVAJJALA
                          PRIMARY EXAMINER